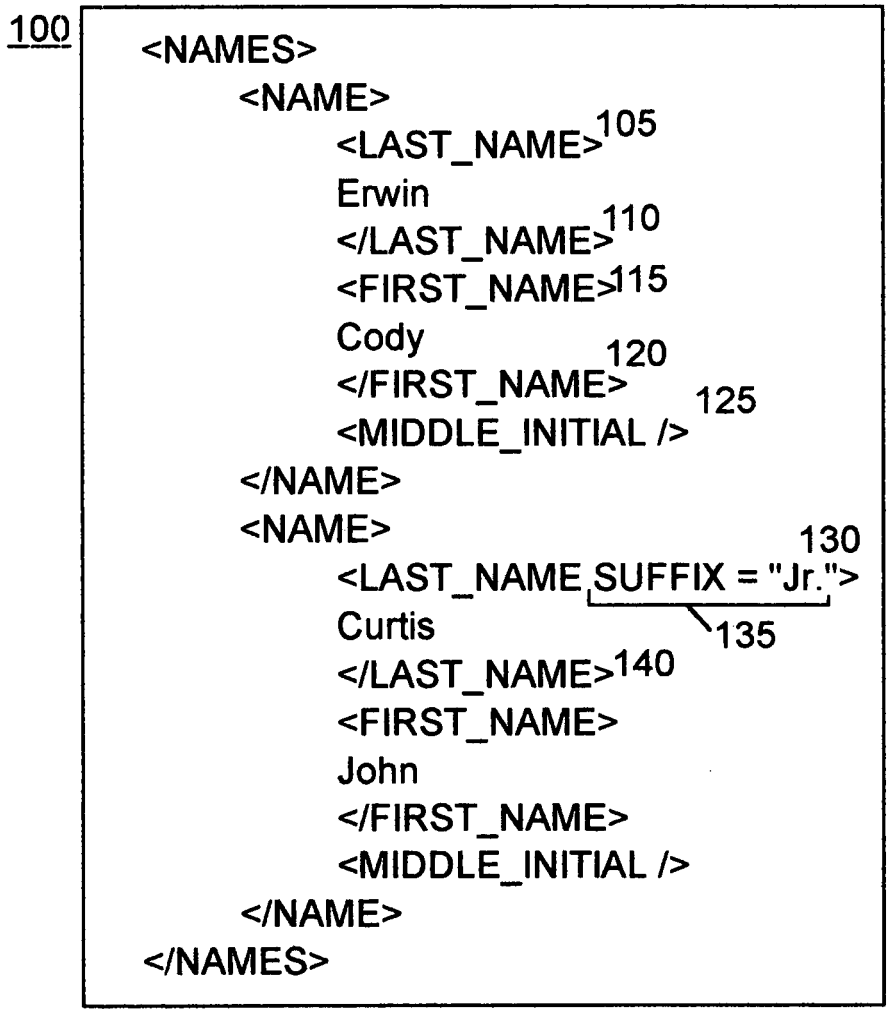
FIG. 1
(PRIOR ART)



0076807 0802596

FIG. 2
(PRIOR ART)

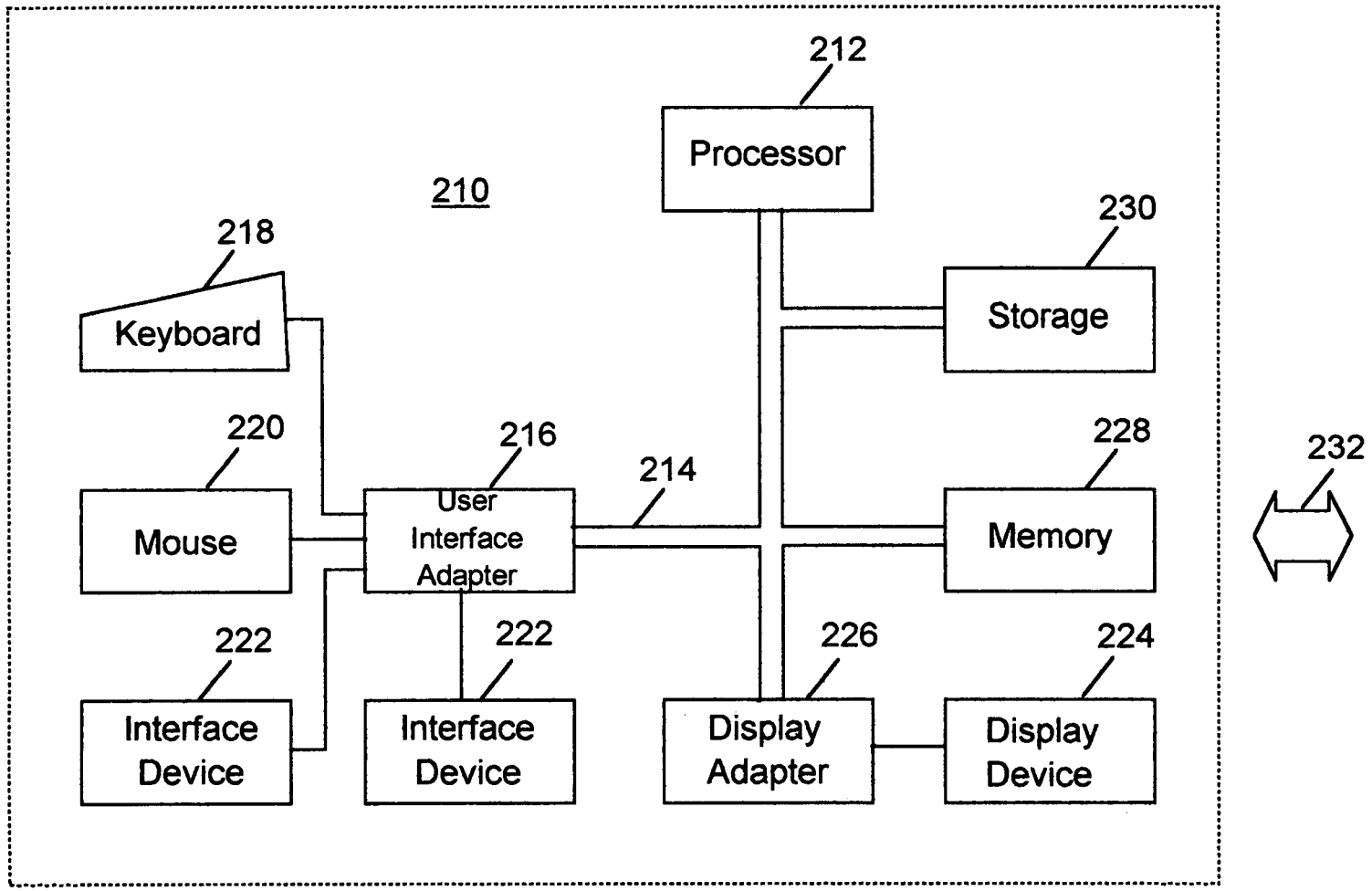
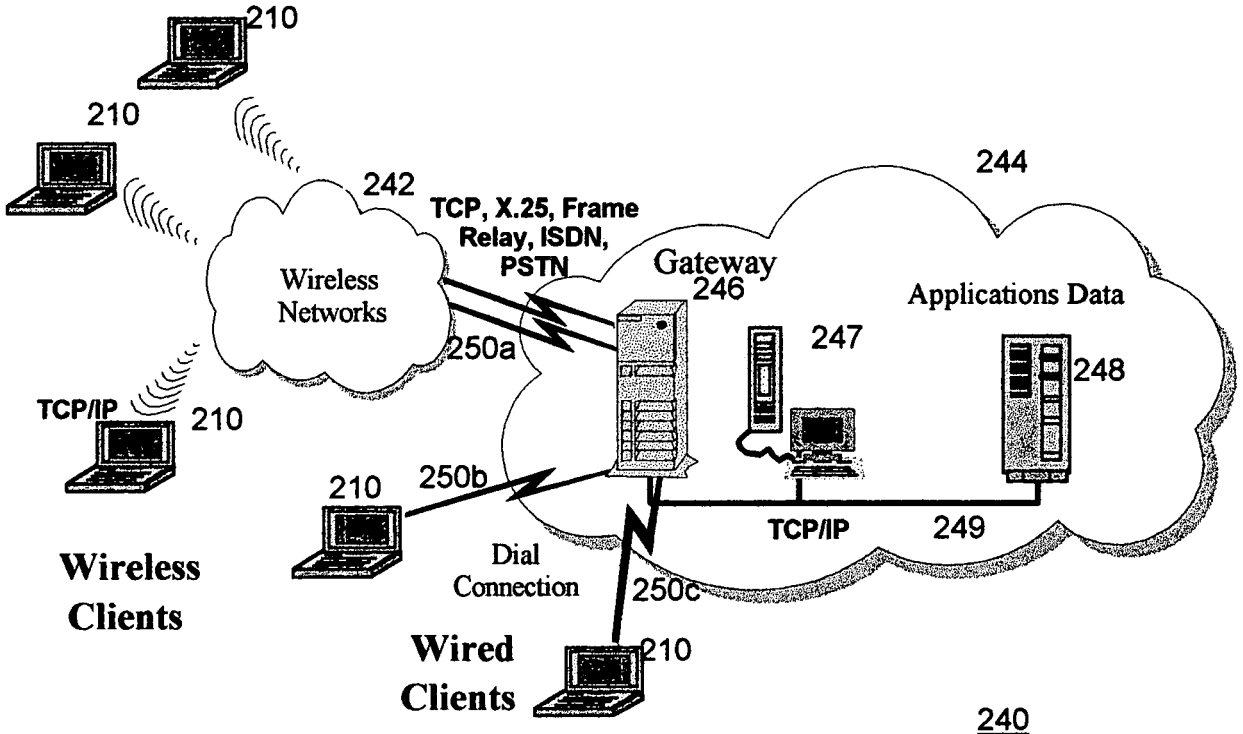


FIG. 3
(PRIOR ART)



00f580"0806e960

FIG. 4A
(PRIOR ART)

400

```
<root_element>                                                    402
    <level_one_element1 id="1" name="1">                          410
        A
        <level_two_element11> B </level_two_element11>            412
        <level_two_element12> C </level_two_element12>            414
    </level_one_element1>
    <level_one_element2 id="2" name="2">                          420
        D
        <level_two_element21> E </level_two_element21>            422
    </level_one_element2>
</root_element>
```

FIG. 4B
(PRIOR ART)

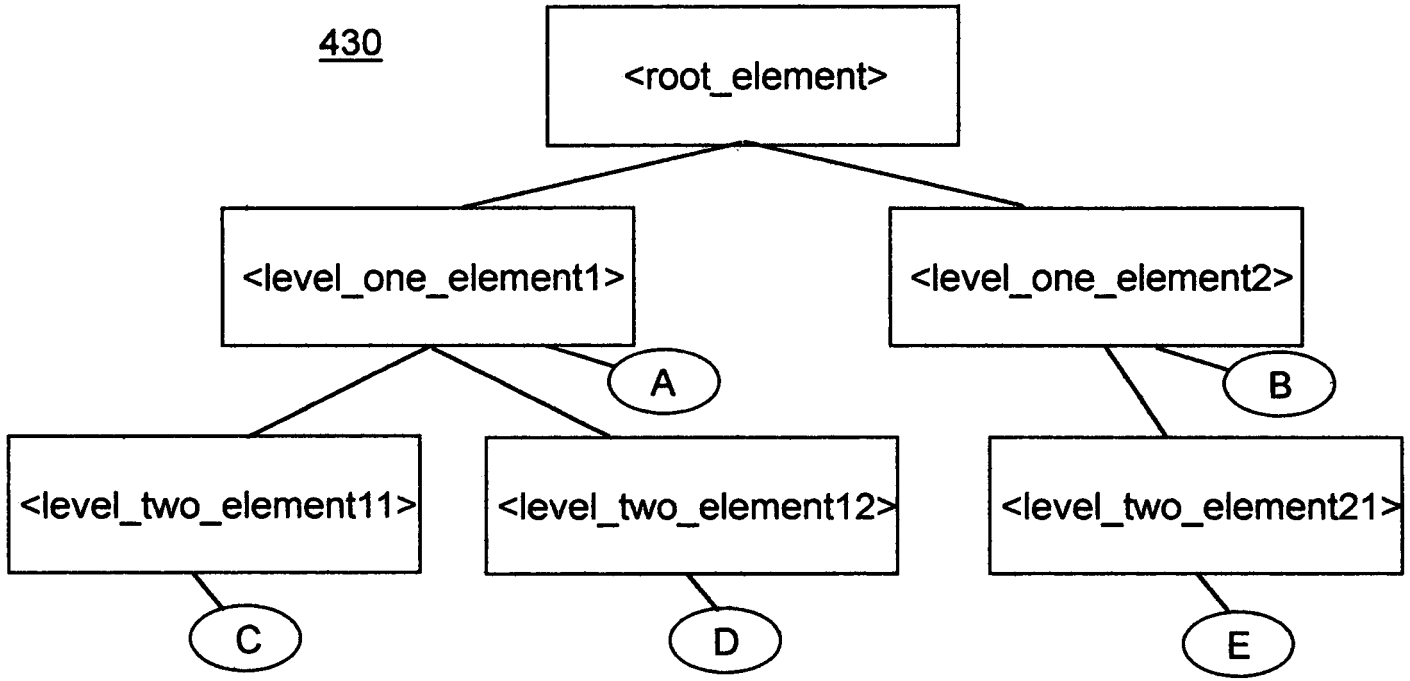


FIG. 4C

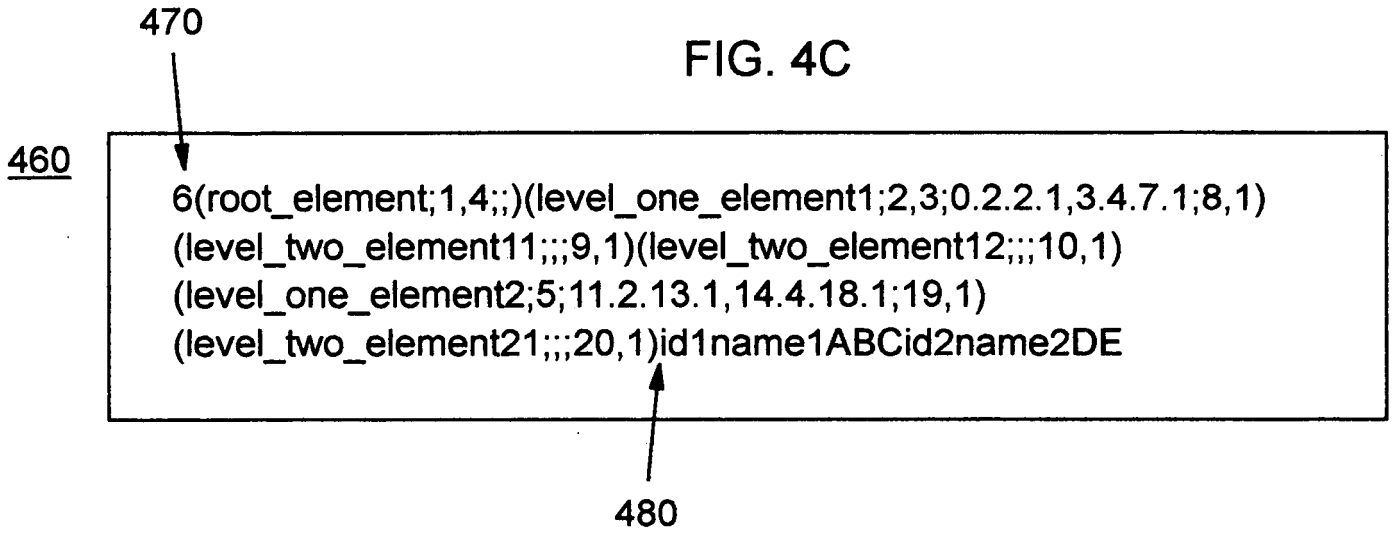

470
460
6(root_element;1,4;;)(level_one_element1;2,3;0.2.2.1,3.4.7.1;8,1)
(level_two_element11;;;9,1)(level_two_element12;;;10,1)
(level_one_element2;5;11.2.13.1,14.4.18.1;19,1)
(level_two_element21;;;20,1)id1name1ABCid2name2DE
480

FIG. 5A

500

	0 505	1 506	2	3 507	4 508	5
501 start pointer	2	22	66	93	121	168
502 length	12	18	19	19	18	19
503 buffer pointer	B	B	B	B	B	B

points to buffer

FIG. 5B

510

	0 515	1	2	3 516	4	5
511 start pointer	-1	8	9	10	19	20
512 length	-1	1	1	1	1	1
513 buffer pointer	DB	DB	DB	DB	DB	DB

points to data buffer

Oracle-Exhibit

FIG. 5C

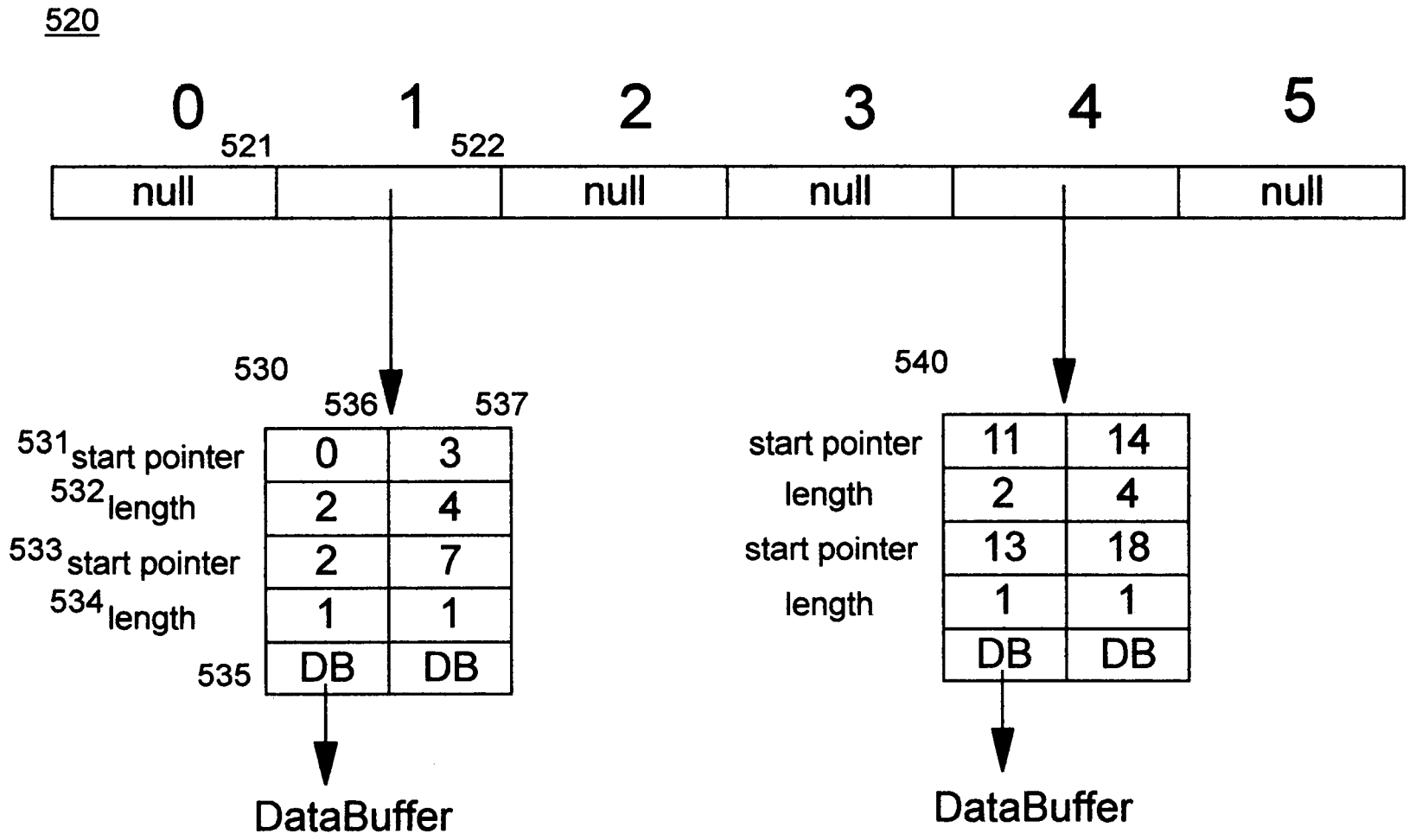


FIG. 5D

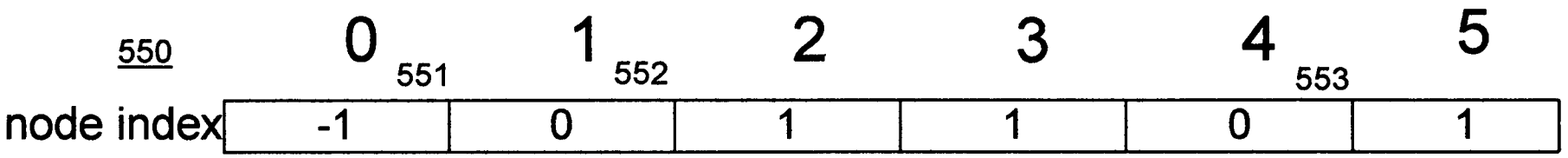

550
0
551
1
552
2
3
4
553
5
node index
-1
0
1
1
0
1

FIG. 5E

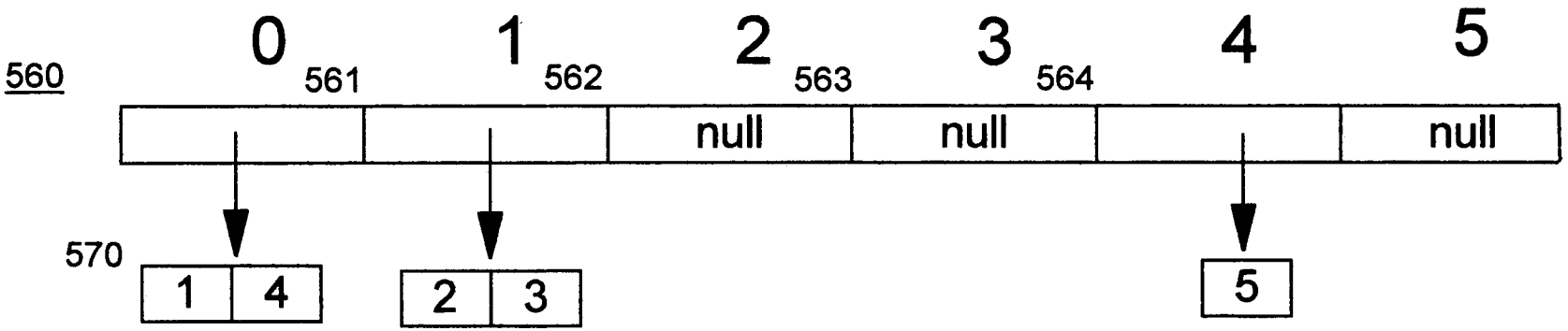
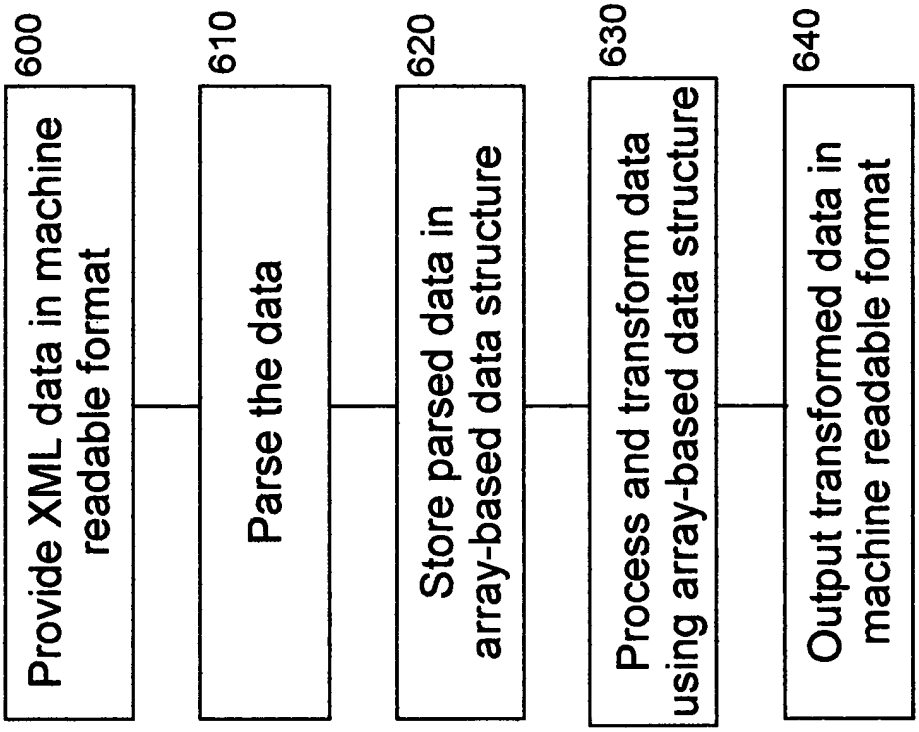

560
0
561
1
562
2
563
3
564
4
5
null
null
null
570
1
4
2
3
5

FIG. 6


600
Provide XML data in machine readable format
610
Parse the data
620
Store parsed data in array-based data structure
630
Process and transform data using array-based data structure
640
Output transformed data in machine readable format

patent application

FIG. 7

700

```
                    705          710
<MAP OPCODE = "RENAME">
    <PARAM>715
        <SNODE>720    725          730
            <XMLDATA TAG = "LAST_NAME" />
        </SNODE> 735
        <TNODE>740    745
            <TAG>SURNAME</TAG>
        </TNODE> 750
    </PARAM>
</MAP>
```

FIG. 8A

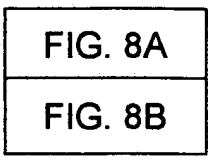


FIG. 8

```
<!ELEMENT operation (map+)>
<!ATTLIST operation
   name CDATA #IMPLIED>

<!ELEMENT map (param+)>
<!ATTLIST map
   opcode (CDATA) #REQUIRED>

<!ELEMENT param (snode, tnode)>

<!ELEMENT snode (xmldata)>
<!ATTLIST snode
   doc CDATA #IMPLIED
   collatekey CDATA #IMPLIED
   pathback CDATA #IMPLIED>

<!ELEMENT xmldata (xml_attribute*)>
<!ATTLIST xmldata
   tag CDATA #IMPLIED
   level CDATA #IMPLIED
   xpathindex CDATA #IMPLIED
   group CDATA #IMPLIED
   invertmatch (true|false) "false">
```

FIG. 8B

```
<!ELEMENT xml_attribute (#PCDATA)>
<!ATTLIST xml_attribute
    name CDATA #REQUIRED
    value CDATA #IMPLIED
    group CDATA #IMPLIED
    invertmatch (true|false) "false">

<!ELEMENT tnode (tag?, attribute*, anchor?)>
<!ATTLIST tnode
    doc CDATA #IMPLIED
    collatekey CDATA #IMPLIED>

<!ELEMENT tag (#PCDATA)>

<!ELEMENT attribute (#PCDATA)>
<!ATTLIST attribute
    name CDATA #REQUIRED
    value CDATA #REQUIRED>

<!ELEMENT anchor (xmldata)>
<!ATTLIST anchor
    rel CDATA #IMPLIED
    place CDATA #IMPLIED>
```

FIG. 9A

900

```
<Operations>
  <operation name="splitOrders">                                    905
    <map opcode="DELETE">                                           910
      <param>
        <snode>
          <xmldata tag="log" level="4" />                           915
        </snode>
        <tnode />
      </map>
      <map opcode="FILTER">                                         920
        <param>
          <snode>
            <xmldata tag="Item" level="4" pathback="2">             925
              <xml_attribute name="description" value="tv-19" group="external" />
              <xml_attribute name="description" value="vcr" group="external" />   930
            </xmldata>
          </snode>
          <tnode doc="externalPOs">                                 935
            <tag>Item</tag>
            <anchor rel="PARENT">                                   940
                                                                    945
              <xmldata tag="externalPurchaseOrders" level="1" />
            </anchor>
          </tnode>
        </param>
        <param>
          <snode pathback="1">
            <xmldata tag="POSummary" level="3" />
          </snode>
          <tnode doc="externalPOs" />
        </param>
```

FIG. 9A
FIG. 9B
FIG. 9

FIG. 9B

```
<param>
 <snode>
  <xmldata tag="Item" level="4">
   <xml_attribute name="description" value="tv-19" invertmatch="true" />
   <xml_attribute name="description" value="vcr" invertmatch="true" />
  </xmldata>
 </snode>
 <tnode doc="internalPOs">
  <tag>Item</tag>
  <anchor rel="PARENT">
   <xmldata tag="internalPurchaseOrders" level="1" />
  </anchor>
 </tnode>
</param>
<param>
 <snode pathback="1">
  <xmldata tag="POSummary" level="3" />
 </snode>
 <tnode doc="internalPOs" />
</param>
</map>
</operation>
```

950

955

960

965

FIG. 10

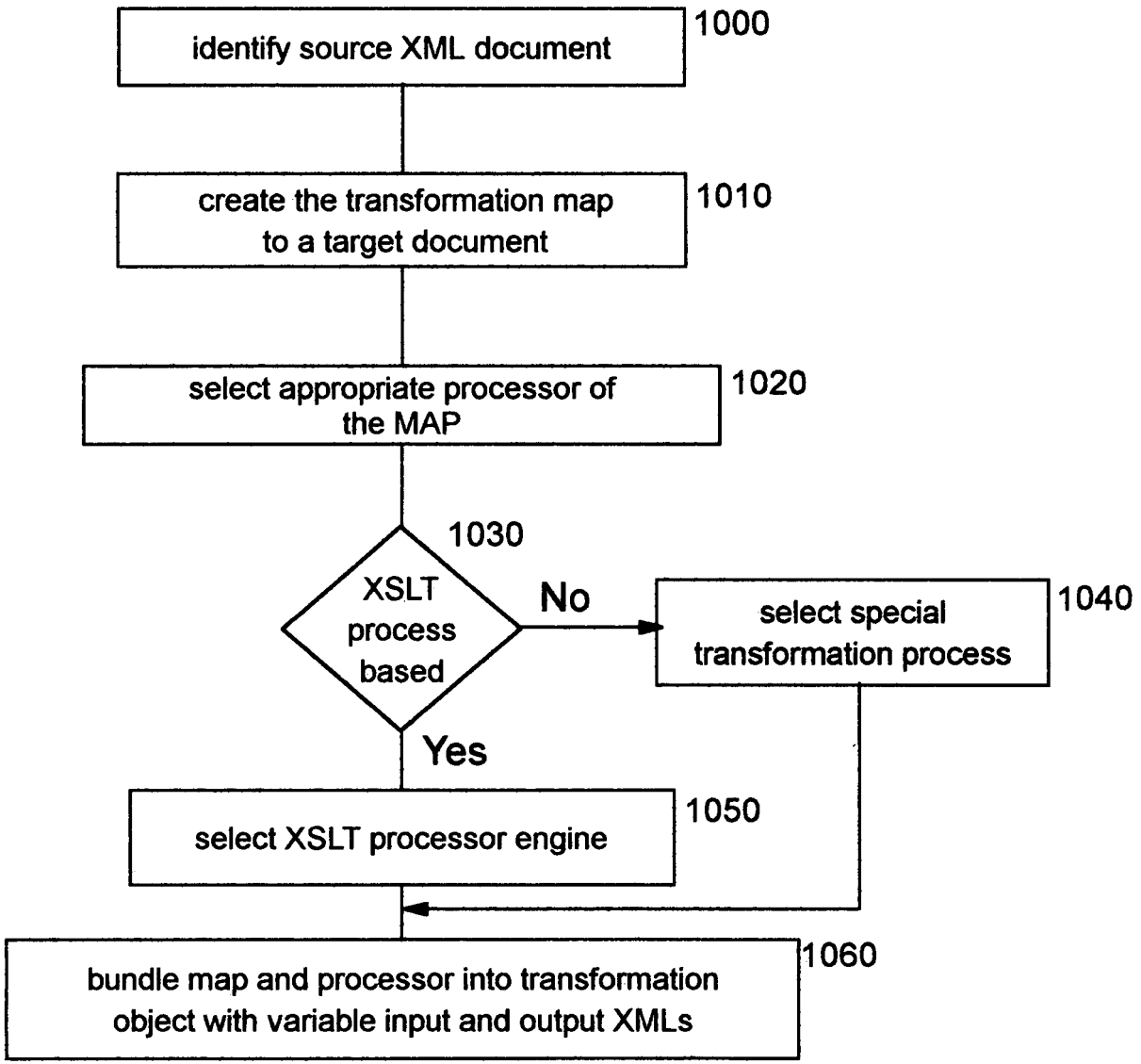

identify source XML document
1000
create the transformation map to a target document
1010
select appropriate processor of the MAP
1020
1030
XSLT process based
No
select special transformation process
1040
Yes
select XSLT processor engine
1050
bundle map and processor into transformation object with variable input and output XMLs
1060

FIG. 11

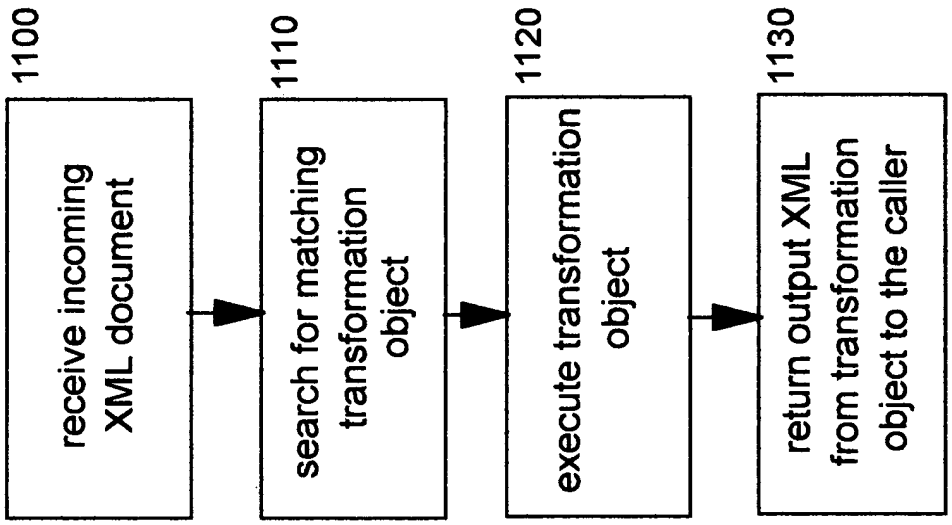

1100
receive incoming XML document
1110
search for matching transformation object
1120
execute transformation object
1130
return output XML from transformation object to the caller

FIG. 12

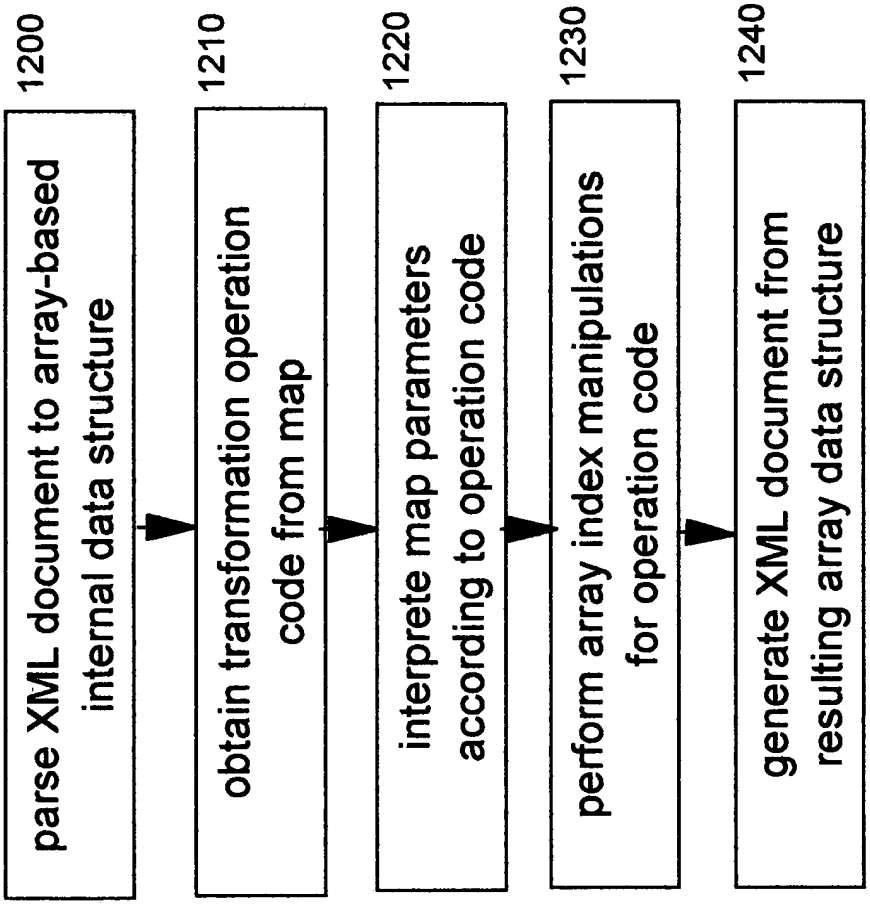

1200
parse XML document to array-based internal data structure
1210
obtain transformation operation code from map
1220
interprete map parameters according to operation code
1230
perform array index manipulations for operation code
1240
generate XML document from resulting array data structure

FIG. 13

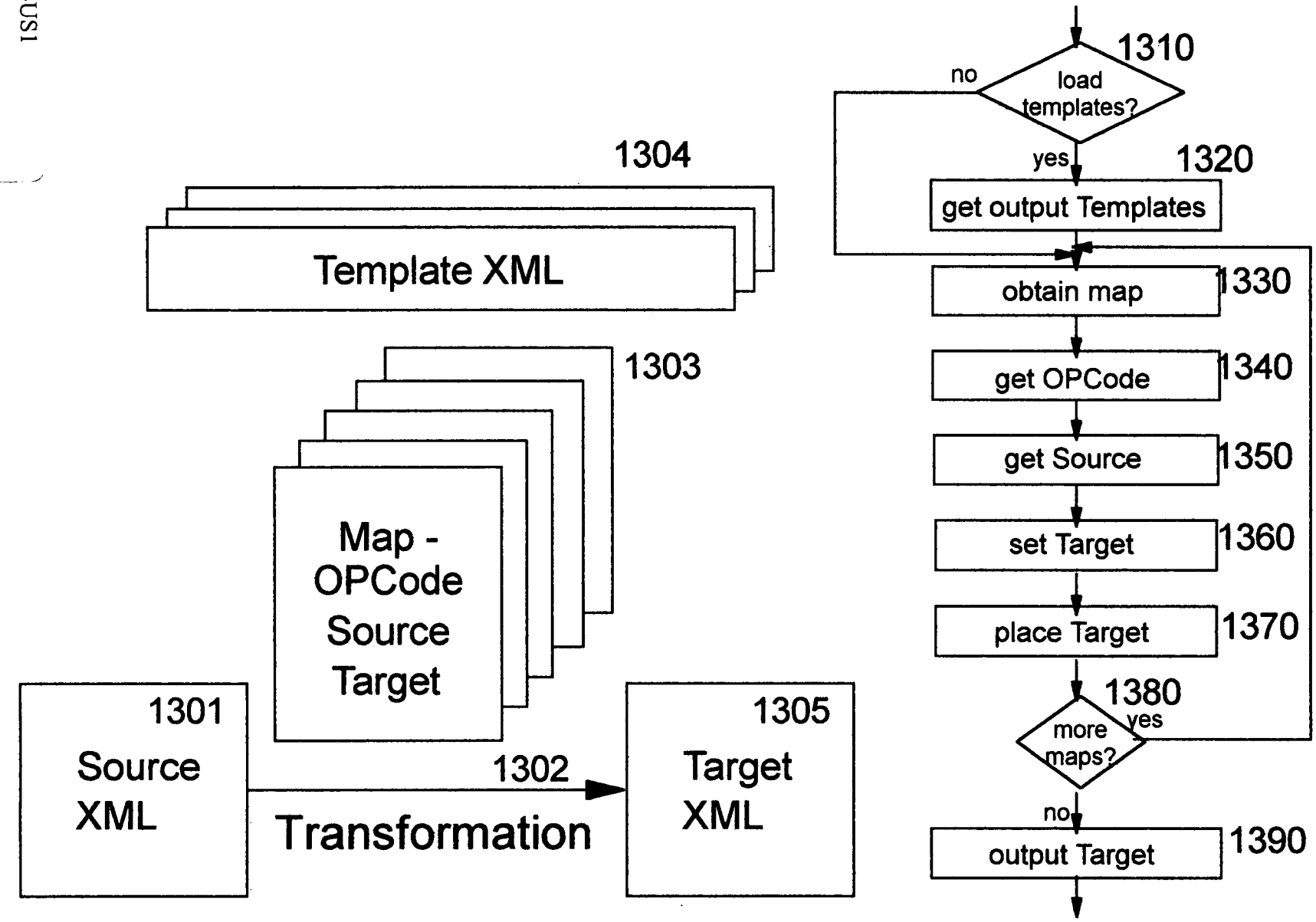

1304
Template XML
1303
Map -
OPCode
Source
Target
1301
Source
XML
1302
Transformation
1305
Target
XML
1310
load
templates?
no
yes
1320
get output Templates
obtain map
1330
get OPCode
1340
get Source
1350
set Target
1360
place Target
1370
1380
more
maps?
yes
no
output Target
1390

FIG. 14A

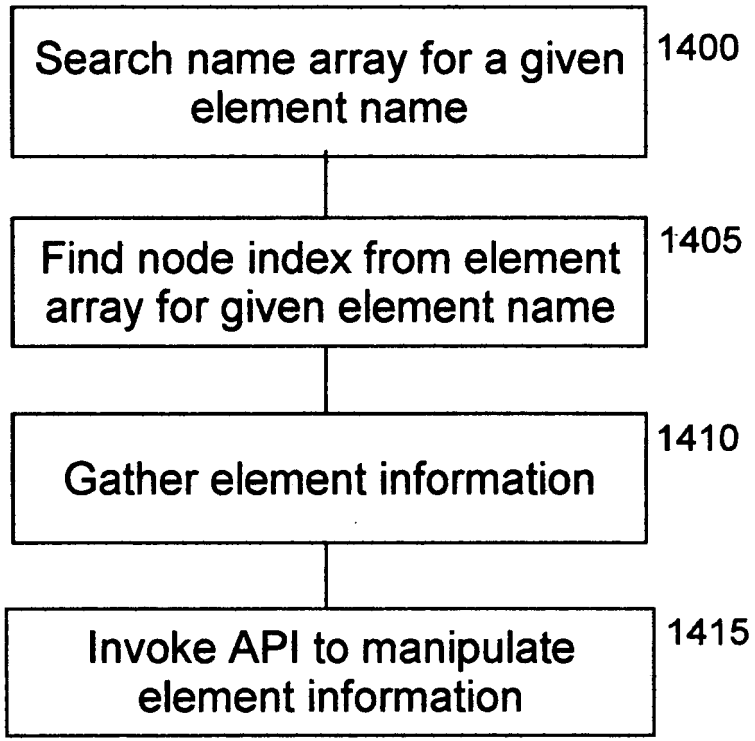

Search name array for a given element name
1400
Find node index from element array for given element name
1405
Gather element information
1410
Invoke API to manipulate element information
1415

FIG. 14B

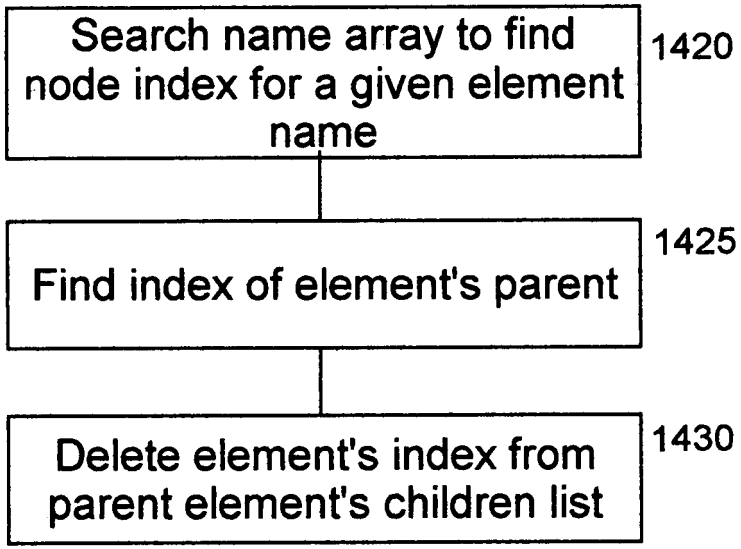


FIG. 14C

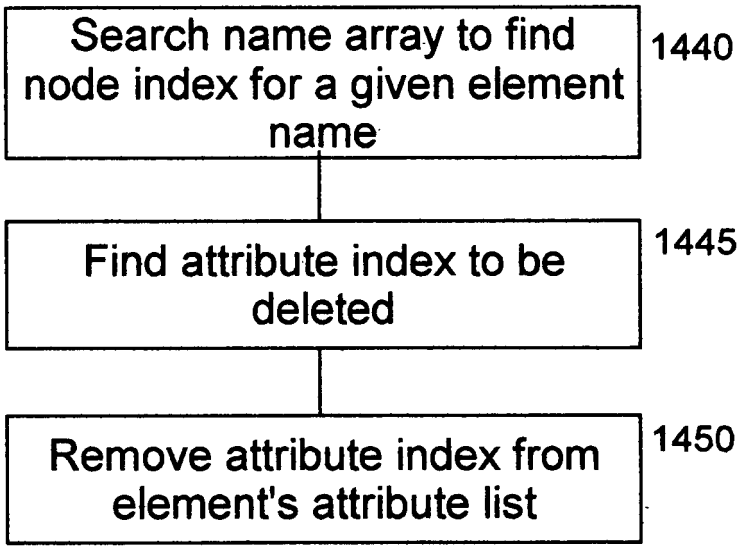
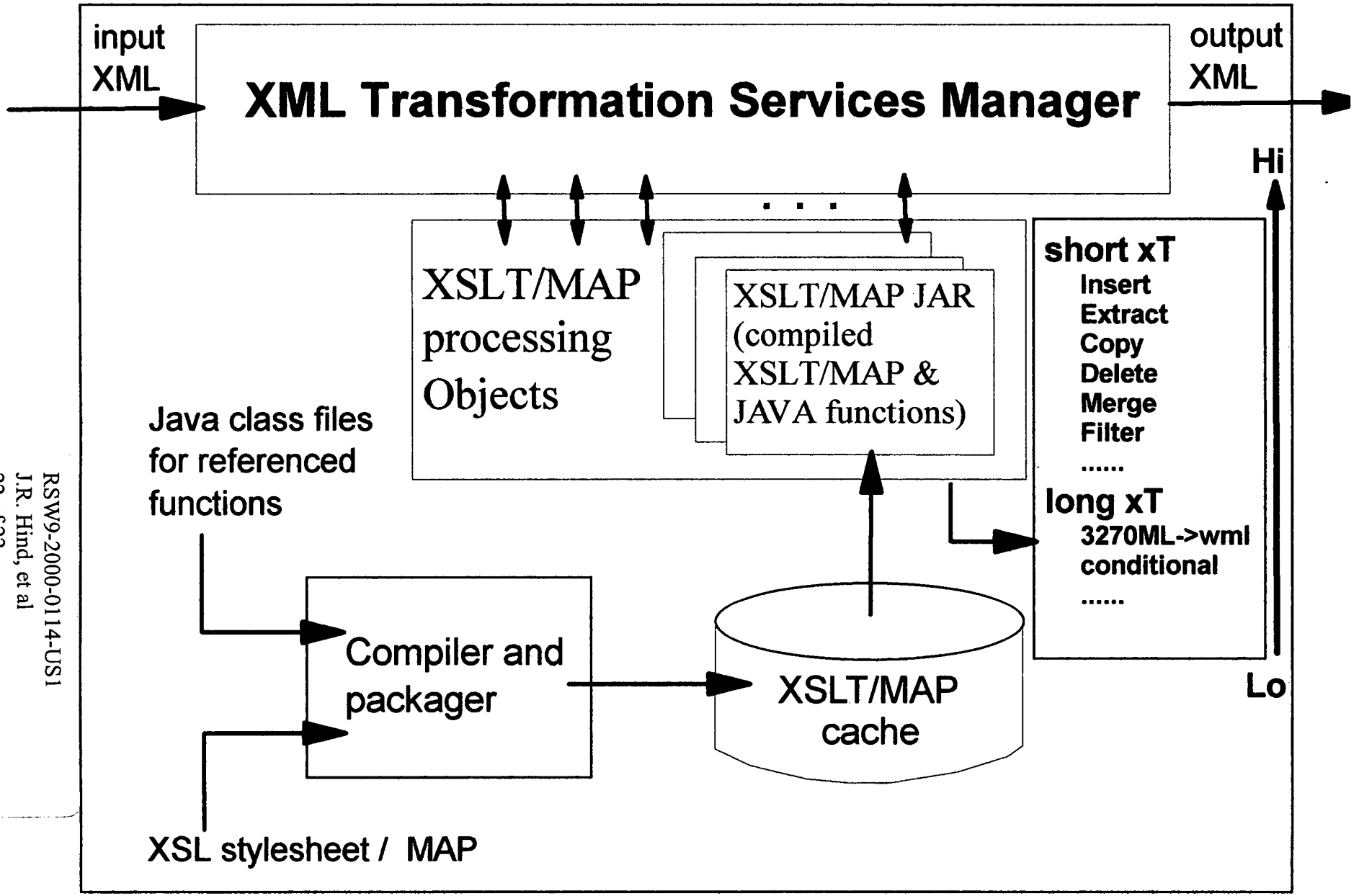


FIG. 15


input XML
XML Transformation Services Manager
output XML
Hi
XSLT/MAP processing Objects
XSLT/MAP JAR (compiled XSLT/MAP & JAVA functions)
short xT
Insert
Extract
Copy
Delete
Merge
Filter
.....
long xT
3270ML->wml
conditional
.....
Java class files for referenced functions
Compiler and packager
XSLT/MAP cache
Lo
XSL stylesheet / MAP